HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200314929-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s):	SHAN, et al.	Confirmation No.:	8714
Application No.:	10/814,715	Examiner:	Harper, L.
Filing Date:	March 31, 2004	Group Art Unit:	2166

Title: LOADING DATA FROM A VERTICAL DATABASE TABLE INTO A HORIZONTAL DATABASE TABLE

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on July 6, 2007.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

☐ (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

☐ 1st Month $120	☐ 2nd Month $450	☐ 3rd Month $1020	☐ 4th Month $1590

☐ The extension fee has already been filed in this application.

☒ (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of $ 500. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees.

☒ A duplicate copy of this transmittal letter is enclosed.

☒ I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to:
Commissioner for Patents, Alexandria, VA 22313-1450
Date of Deposit: September 6, 2007

OR

☐ I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (571)273-8300.
Date of facsimile:
Typed Name: Kelly Davis
Signature: Kelly Davis

Respectfully submitted,
SHAN, et al.
By _____
LeRoy D. Maunu
Attorney/Agent for Applicant(s)
Reg No. : 35,274
Date : September 6 , 2007
Telephone : 651-686-6633

Rev 10/06a (AplBrief)

09/10/2007 CNEGA1 00000013 082025 10814715
01 FC:1402 500.00 DA

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	SHAN et al.	Examiner:	Harper, L.
Serial No.:	10/814,715	Group Art Unit:	2166
Filed:	March 31, 2004	Docket No.:	200314929-1 (HPCO.144PA)
Title:	LOADING DATA FROM A VERTICAL DATABASE TABLE INTO A HORIZONTAL DATABASE TABLE		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Board of Patent Appeals and Interferences, United States Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450, September 6, 2007.

By: Kelly Davis
Kelly Davis

Board of Patent Appeals and Interferences
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is an Appeal Brief submitted pursuant to 37 C.F.R. § 41.37 for the above-referenced patent application.

I. Real Party in Interest

The real party in interest is Hewlett-Packard Development Company, L.P., having its principal place of business Houston, Texas. The above referenced patent application is assigned to Hewlett-Packard Company, L.P.

II. Related Appeals and Interferences

Appellant is unaware of any related appeals, interferences or judicial proceedings.

III. Status of Claims

Claims 1, 8, 18, and 25 are rejected, and claims 2-7, 9-17, 19-24, and 26-30 are objected to. The Examiner indicated that claims 2-7, 9-17, 19-24, and 26-30 would be allowable if rewritten to include the limitations of any intervening claims. The appealed claims are in the attached Appendix of Appealed Claims.

IV. Status of Amendments

No amendment after final rejection was filed, and all amendments filed before the final rejection have been entered.

V. Summary of Claimed Subject Matter

In the embodiment set forth in claim 1, the invention provides a processor-implemented method for loading data from a vertical database table into a horizontal database table. The method comprises determining a number of rows in a vertical database table (FIG. 3, 304; [0026]) and a number of columns in a horizontal database table (FIG. 3, 302; [0024]). The vertical table includes at least three columns ([0005]), with entries in a first column containing object identifiers, entries in a second column containing attribute names corresponding to the object identifiers, and entries in a third column containing attribute values corresponding to the attribute names (e.g., FIG. 2, 202; [0005]). The horizontal table includes a column for the object identifiers and attribute columns for the attributes names (e.g., FIG. 2, 204; [0007]). The method further comprises selecting one of a plurality of methods (FIG. 2, 310, 312, 320, 322, 326, 332, 334; [0015], [0017], [0023]) for reading data from the vertical database table and writing data to the horizontal database table based in part on the number of rows in the vertical database table (FIG. 2, 308, 314, 318, 330; [0027]-[0034]) and the number of columns in the horizontal database table (FIG. 2, 306; [0027]-[0034]). The method reads object identifiers and values of attributes from the vertical database table (FIG. 4, 406; e.g., [0038]; FIG. 5, 506; e.g., [0043]) and writes the object identifiers and the values of attributes to the vertical database table using the selected one of the plurality of methods (FIG. 4, 416, 418; [0041]-[0042]; FIG. 5, 516; [0044]-[0047]).

Another embodiment, as found in claim 18, sets forth a program storage medium. The program storage medium comprises at least one processor-readable program storage device configured with instructions for loading data from a vertical database table into a horizontal database table. The execution of the instructions by one or more processors causes the one or more processors ([0048]) to perform operations including, determining a number of rows in a vertical database table (FIG. 3, 304; [0026]) and a number of columns in a horizontal database table (FIG. 3, 302; [0024]). The vertical table includes at least three columns ([0005]), with entries in a first column containing object identifiers, entries in a second column containing attribute names corresponding to the object identifiers, and entries in a third column containing attribute values corresponding to the attribute names (e.g., FIG. 2, 202; [0005]). The horizontal table includes a column for the object identifiers and respective attribute columns for the attributes names (e.g., FIG. 2, 204; [0007]). A further operation performed by the one or more processors is selecting one of a plurality of methods (FIG. 2, 310, 312, 320, 322, 326, 332, 334; [0015], [0017], [0023]) for reading data from the vertical database table and writing data to the horizontal database table based on the number of rows in the vertical database table (FIG. 2, 308, 314, 318, 330; [0027]-[0034]) and the number of columns in the horizontal database table (FIG. 2, 306; [0027]-[0034]). Further operations performed by the one or more processors include reading object identifiers and attributes values from the vertical database table (FIG. 4, 406; e.g., [0038]; FIG. 5, 506; e.g., [0043]) and writing the object identifiers and attributes values to the vertical database table using the selected one of the plurality of methods (FIG. 4, 416, 418; [0041]-[0042]; FIG. 5, 516; [0044]-[0047]).

VI. Grounds of Rejection

Claims 1, 8, 18, and 25 stand rejected under 35 U.S.C. §102(e) as being anticipated by "Agrawal" (U.S. Patent 6,763,350 to Agrawal et al.).

VII. Argument

The rejection of claims 1, 8, 18, and 25 should be reversed because the Examiner has not shown that the claims are anticipated under 35 U.S.C. §102(e) by Agrawal.

Claims 1, 8, 18, and 25

The rejection of claims 1, 8, 18, and 25 under 35 USC §102(e) should be reversed because the Examiner has not shown that all the limitations of the claims are taught by Agrawal.

According to claim 1, the limitations include selecting one of a plurality of methods for reading data from a vertical database table and writing data to the horizontal database table based in part on the number of rows in the vertical database table and the number of columns in the horizontal database table; and reading object identifiers and values of attributes from the vertical database table and writing the object identifiers and the values of attributes to the vertical database table using the selected one of the plurality of methods. The Examiner has failed to show that Agrawal teaches all of these limitations.

Neither the cited portions of Agrawal nor other un-cited portions appear to teach selecting of one of a plurality of methods based in part on the number of rows in the vertical database table and the number of columns in the horizontal database table. Agrawal generally teaches defining a horizontal view over an underlying vertical table and then for queries posed against this view transforming and executing the queries against the vertical tables (Abstract).

In the cited col. 5, l. 5-30, Agrawal apparently teaches a single vertical-to-horizontal transformation ("v2h") that is apparently denoted with: Ω. The $\Omega^k(V)$ function creates a horizontal table of arity k+1 from a vertical table, V. Note that the h2v operation converts a horizontal table to a vertical table and is inapplicable to the claim limitations (col. 6, l. 3-9). Thus, there appears to be only a single transformation method, and there is no apparent need for selection based on the number of rows in the vertical database table and the number of columns in the

horizontal database table. Therefore, the cited portions of Agrawal do not suggest the claim limitations.

In another portion of Agrawal, which is not cited, Agrawal suggests "two transformation strategies," "VerticalSQL" and "VerticalUDF," for transforming an SQL query against a horizontal table and transforming the query into an SQL query for the underlying vertical table (col. 10, l. 34 – col. 11, l. 61). Although Agrawal suggests two strategies, neither of the strategies involves selecting one of a plurality of methods for reading data from a vertical database table and writing data to the horizontal database table based in part on the number of rows in the vertical database table and the number of columns in the horizontal database table. The VerticalSQL strategy transforms an input SQL that specifies a horizontal table, "H," into an SQL translation that includes a left outer join directed to the underlying vertical table, "V" (col. 10, lines 40-65). Thus, the VerticalSQL strategy does not relate to selecting one of a plurality of methods for reading data from a vertical database table and writing data to the horizontal database table based in part on the number of rows in the vertical database table and the number of columns in the horizontal database table.

Agrawal's VerticalUDF strategy reads from the vertical table and outputs horizontal tuples. The VerticalUDF strategy essentially takes an input query directed to a horizontal table H and transforms the query into one applicable to the underlying vertical table V (col. 11, lines 12-30). Thus, Agrawal's VerticalUDF strategy does not suggest that there are a plurality of methods to be selected from, and the selection being based in part on the number of rows in the vertical database table and the number of columns in the horizontal database table.

Since neither the cited portions nor other un-cited portions of Agrawal suggest the claim limitations of selecting one of a plurality of methods based in part on the number of rows in the vertical database table and the number of columns in the horizontal database table, the rejection is unsupported and should be reversed.

The Examiner's response to Applicants' arguments is not supported by Agrawal's teachings. The Examiner asserts in his response to Applicants' arguments that one skilled in the art would infer from Agrawal's disclosure that "there

... are a plurality of ways to convert a vertical database into a horizontal database and this is in fact based on the number of rows in the database." However, based on the cited teachings of Agrawal one would not infer a plurality of ways to convert a vertical table into a horizontal table.

The Examiner gives two reasons to support this assertion and draws on teachings from Agrawal to support these reasons. The cited portions of Agrawal do not support the Examiner's view.

The first reason given by the Examiner to support his assertion is that "as the disclosure states sqlis [sic] designed for horizontal formats (See column 1 lines 64-66) and when horizontal databases become too large, there are many nulls that have to be placed in the database (See column 1 lines 33-39)." Taken in the full context of Agrawl's Background, however, one would not reasonably infer from these cited aspects of Agrawal, that there is a plurality of ways to convert a vertical table into a horizontal table. Nor has the Examiner explained how one skilled in the art would draw the inference of multiple ways to convert a vertical table into a horizontal table.

Agrawal's Background explains that SQL was designed for horizontal databases (col. 1, lines 15-20). Horizontal databases may have many columns, and in some instances many of those columns would contain nulls (the database would be sparse) (col. 1, lines 23-52). Vertical databases alleviate the problem of sparseness (col. 1, lines 53-63). However, SQL queries against vertical databases require a great level of expertise (col. 1, line 64 – col. 2, line 6). Thus, Agrawal's disclosure teaches ways to support querying a vertical database using conventional SQL. Agrawal defines a horizontal view over the underlying vertical tables and the queries are posed against this view, with the queries being automatically transformed and executed against the vertical tables (Abstract). Thus, Agrawal's Background does not imply that there are multiple methods to select from for reading data from a vertical database table and writing data to the horizontal database table based in part on the number of rows in the vertical database table and the number of columns in the horizontal database table.

The second reason given by the Examiner to support his assertion is that "the disclosure of the v2h function is the preferred embodiment of ... [Agrawal's] invention, and any artisan of ordinary skill in the pertinent art when looking at claims 3 and 4 (keeping in mind the concept of claim differentiation) would see that it is unequivocally clear that a plurality of functions to convert vertical databases to horizontal databases were disclosed." However, when claim 5 is considered along with claims 3 and 4, there is no apparent inference of multiple functions to convert a vertical database into horizontal database.

Agrawal's claim 3 depends from claim 2 which depends from claim 1. Claim 1 sets forth a step of "transforming the query to render a transformed query." Claim 2 sets forth that "the query is transformed using at least one operator." Claim 3 sets forth that "the operator receives at least one vertical table ... and outputs the logical horizontal table." Claim 4 depends from claim 3 and sets forth that "the operator is a v2h operator." Claim 5 also depends from claim 3 and sets forth that "the operator executes a left outer join of a projection ... of identifiers of the vertical table with a sequence of left outer joins of a set of projections of attribute values from the vertical table."

From Agrawl's claims it can be seen that the "operator" may be either a v2h operator (claim 4) or a left outer join (claim 5). As explained above, Agrawal teaches two transformation strategies, and the v2h of claim 4 corresponds to the VerticalUDF strategy, and the left outer join of claim 5 corresponds to the VerticalSQL strategy (col. 10, line 39 – col. 11, line 22). Thus, Agrawal's left outer join is not a type of v2h operator. Nor that there are multiple v2h approaches. Rather, Agrawal's claims set forth two alternatives for the operator of claim 3. One being a v2h operator and the other being a left outer join. Thus, the Examiner's assertion that Agrawal's "claim 3 must cover more than just the single v2h operator" is not supported by Agrawal's disclosure and claims. Rather, Agrawal's "operator" in claim 3 is understood to cover either the v2h that reads tuples from a vertical table and outputs horizontal tuples (col. 11, lines 3-11), or a left outer join that is a translation of the input query (col. 10, lines 41-67). Thus, Agrawal does not implicitly disclose multiple methods to select from for reading data from a vertical database table and writing data to the

horizontal database table based in part on the number of rows in the vertical database table and the number of columns in the horizontal database table.

Even if one assumes for purposes of discussion that Agrawal implicitly discloses multiple ways to convert a vertical table into a horizontal table, the Examiner still has not shown where the selection of one of these "implicit" ways is based in part on the number of rows in the vertical database table and the number of columns in the horizontal database table. Thus, the Examiner has not shown that Agrawal anticipates all the limitations of claim 1, and the rejection should be reversed.

Claim 18 is directed to a program storage medium and includes functional limitations similar to those of claim 1 as discussed above. Claim 8 depends from claim 1, and claim 25 depends from claim 18. Thus, the limitations of claims 8, 18, and 25 are not shown to be taught by Agrawal for at least the reasons set forth above.

Claims 1, 8, 18, and 25 are not anticipated by Agrawal, and Applicant respectfully requests reversal of the rejection of these claims.

Claims 2-7, 9-17, 19-24, and 26-30

Claims 2-7, 9-17, 19-24, and 26-30 are separately patentable over claims 1, 8, 18, and 25. Having been deemed to be allowable if rewritten to include the limitations of intervening claims, claims 2-7, 9-17, 19-24, and 26-30 are by definition separately patentable over claims 1, 8, 18, and 25.

VIII. Conclusion

In view of the above, Appellant submits that the rejection and objection are improper, and the claimed invention is patentable. Appellant respectfully requests reversal of the rejection of claims 1, 8, 18, and 25, reversal of the objection to claims 2-7, 9-17, 19-24, and 26-30, and allowance of the entire application.

Respectfully submitted,

By: _____

Name: LeRoy D. Maunu
Reg. No.: 35,274

APPENDIX OF APPEALED CLAIMS FOR APPLICATION NO. 10/814,715

1. A processor-implemented method for loading data from a vertical database table into a horizontal database table, comprising:

determining a number of rows in a vertical database table and a number of columns in a horizontal database table, wherein the vertical table includes at least three columns, with entries in a first column containing object identifiers, entries in a second column containing attribute names corresponding to the object identifiers, and entries in a third column containing attribute values corresponding to the attribute names, and the horizontal table includes a column for the object identifiers and attribute columns for the attributes names;

selecting one of a plurality of methods for reading data from the vertical database table and writing data to the horizontal database table based in part on the number of rows in the vertical database table and the number of columns in the horizontal database table; and

reading object identifiers and values of attributes from the vertical database table and writing the object identifiers and the values of attributes to the vertical database table using the selected one of the plurality of methods.

2. The method of claim 1, wherein access to the horizontal database table is provided by a database management system, the method further comprising:

in response to selection of a first one of the plurality of methods,

for each object identifier in the file,

generating a character string array having in character string format, data from the vertical database table including, an object identifier and attribute values for attributes associated with the object identifier; and

writing data from the character string array to a flat text file; and

loading data from the flat text file into the horizontal database using a utility function of the database management system, wherein the utility function is adapted to read data from a flat text file.

3. The method of claim 2, wherein generating a character string array includes determining a position for storage of an attribute value in the character string array using a hash of the attribute name associated with the attribute value.

4. The method of claim 2, wherein the first one of the plurality of methods is selected in response to the number of rows in the vertical table being greater than a first threshold and the number of columns in the horizontal table being less than a second threshold.

5. The method of claim 1, wherein access to the horizontal database table is provided by a database management system, the method further comprising:

in response to selection of a first one of the plurality of methods,

for each object identifier in the file,

generating a character string array having in character string format, data from the vertical database table including, an object identifier and attribute values for attributes associated with the object identifier;

generating an SQL insert command from the contents of the character string array;

issuing the SQL insert command to the database management system; and

issuing an SQL commit command to the database management system.

6. The method of claim 5, wherein generating a character string array includes determining a position for storage of an attribute value in the character string array using a hash of the attribute name associated with the attribute value.

7. The method of claim 5, wherein the first one of the plurality of methods is selected in response to the number of rows in the vertical table being less than a first threshold and the number of columns in the horizontal table being less than a second threshold.

8. The method of claim 1, wherein access to the vertical database table and access to the horizontal database table are provided by a database management system, the method further comprising:

in response to selection of a first one of the plurality of methods,

generating a single SQL command that selects data from the vertical table and inserts the data in the horizontal table for each object identifier in the vertical table and each column in the horizontal table; and

issuing the SQL command to the database management system.

9. The method of claim 8, further comprising:

determining an approximate percentage of a total number of attribute values that will have an undefined value in the horizontal table;

wherein the first one of the plurality of methods is selected in response to the approximate percentage being less than a first threshold percentage, the number of columns in the horizontal table being less than a second threshold, and the number of rows in the vertical table being less than a third threshold.

10. The method of claim 9, further comprising:

in response to selection of a second one of the plurality of methods,

for each object identifier in the file,

generating a character string array having in character string format, data from the vertical database table including, an object identifier and attribute values for attributes associated with the object identifier; and

writing data from the character string array to a flat text file; and

loading data from the flat text file into the horizontal database using a utility function of the database management system, wherein the utility function is adapted to read data from a flat text file; and

wherein the second one of the plurality of methods is selected in response to the number of rows in the vertical table being greater than a fourth threshold and the number of columns in the horizontal table being less than the second threshold; and

wherein the second one of the plurality of methods is selected in response to the number of columns in the horizontal table being less than the second threshold, the number of rows in the vertical table being in the range of a fifth to the third threshold, and the approximate percentage being greater than the first threshold percentage.

11. The method of claim 1, wherein access to the vertical database table and access to the horizontal database table are provided by a database management system, the method further comprising:

in response to selection of a first one of the plurality of methods,

for each object identifier in the file,

generating a character string array having in character string format, data from the vertical database table including, an object identifier and attribute values for attributes associated with the object identifier; and

writing data from the character string array to a flat text file; and

loading data form the flat text file into the horizontal database using a utility function of the database management system, wherein the utility function is adapted to read data from a flat text file;

in response to selection of a second one of the plurality of methods,

for each object identifier in the file,

generating a character string array having in character string format, data from the vertical database table including, an object identifier and attribute values for attributes associated with the object identifier;

generating an SQL insert command from the contents of the character string array;

issuing the SQL insert command to the database management system; and

issuing an SQL commit command to the database management system;

in response to selection of a third first one of the plurality of methods,

generating a single SQL command that selects data from the vertical table and inserts the data in the horizontal table for each object identifier in the vertical table and each column in the horizontal table; and

issuing the SQL command to the database management system.

12. The method of claim 11, further comprising:

determining storage space requirements of the flat text file;

determining availability of sufficient retentive storage space based on the storage space requirements of for the flat text file;

selecting the first method in response to the number of columns being less than a first threshold, the number of rows being greater than a second threshold, and there being available sufficient retentive storage space for the flat text file.

13. The method of claim 12, further comprising:

determining an approximate percentage of a total number of attribute values that will have an undefined value in the horizontal table;

in response insufficient retentive storage space being available for the flat text file,

selecting the second method in response to the number of rows being less than a third threshold or the approximate percentage being greater than a fourth threshold percentage;

selecting the third method in response to the number of rows being greater than or equal to the third threshold and the approximate percentage being less than or equal to the fourth threshold percentage.

14. An apparatus for loading data from a vertical database table into a horizontal database table, comprising:

means for determining a number of rows in a vertical database table and a number of columns in a horizontal database table, wherein the vertical table includes at least three columns, with entries in a first column containing object identifiers, entries in a second column containing attribute names corresponding to the object identifiers, and entries in a third column containing attribute values corresponding to the attribute names, and the horizontal table includes a column for the object identifiers and respective attribute columns for the attributes names;

means, responsive to the number of rows in the vertical database table and the number of columns in the horizontal database table, for selecting one of a plurality of approaches for reading data from the vertical database table and writing data to the horizontal database table; and

means for performing the selected one of the plurality of approaches.

15. The apparatus of claim 14, further comprising:

a database management system coupled to the horizontal database table;

means, responsive to a selection of a first one of the plurality of approaches, for generating, for each object identifier in the file, a character string array having in character string format, data from the vertical database table including, an object identifier and attribute values for attributes associated with the object identifier; and

means, responsive to selection of a first one of the plurality of approaches, for writing, for each object identifier in the file, data from the character string array to a flat text file; and

means, responsive to selection of a first one of the plurality of approaches, for loading data from the flat text file into the horizontal database using a utility function of the database management system, wherein the utility function is adapted to read data from a flat text file.

16. The apparatus of claim 14, further comprising:

a database management system coupled to the horizontal database table;

means, responsive to selection of a first one of the plurality of approaches, for generating, for each object identifier in the file, a character string array having in character string format, data from the vertical database table including, an object identifier and attribute values for attributes associated with the object identifier;

means for generating an SQL insert command from the contents of the character string array;

means for issuing the SQL insert command to the database management system; and

means for issuing an SQL commit command to the database management system.

17. The apparatus of claim 14, further comprising:

a database management system coupled to the vertical database table and to the horizontal database table;

means, responsive to selection of a first one of the plurality of approaches, for generating a single SQL command that selects data from the vertical table and inserts the data in the horizontal table for each object identifier in the vertical table and each column in the horizontal table; and

means for issuing the SQL command to the database management system.

18. A program storage medium, comprising:

at least one processor-readable program storage device configured with instructions for loading data from a vertical database table into a horizontal database table, wherein execution of the instructions by one or more processors causes the one or more processors to perform the operations including,

determining a number of rows in a vertical database table and a number of columns in a horizontal database table, wherein the vertical table includes at least three columns, with entries in a first column containing object identifiers, entries in a second column containing attribute names corresponding to the object identifiers, and entries in a third column

containing attribute values corresponding to the attribute names, and the horizontal table includes a column for the object identifiers and respective attribute columns for the attributes names;

selecting one of a plurality of methods for reading data from the vertical database table and writing data to the horizontal database table based on the number of rows in the vertical database table and the number of columns in the horizontal database table; and

reading object identifiers and attributes values from the vertical database table and writing the object identifiers and attributes values to the vertical database table using the selected one of the plurality of methods.

19. The program storage medium of claim 18, wherein access to the horizontal database table is provided by a database management system, and the at least one processor-readable program storage device is further configured with instructions for execution by the one or more processors for performing the operations comprising:

in response to selection of a first one of the plurality of methods,

for each object identifier in the file,

generating a character string array having in character string format, data from the vertical database table including, an object identifier and attribute values for attributes associated with the object identifier; and

writing data from the character string array to a flat text file; and

loading data from the flat text file into the horizontal database using a utility function of the database management system, wherein the utility function is adapted to read data from a flat text file.

20. The program storage medium of claim 19, wherein the instructions for generating a character string array include instructions for determining a position for storage of an attribute value in the character string array using a hash of the attribute name associated with the attribute value.

21. The program storage medium of claim 19, wherein the first one of the plurality of methods is selected in response to the number of rows in the vertical table being greater than a first threshold and the number of columns in the horizontal table being less than a second threshold.

22. The program storage medium of claim 18, wherein access to the horizontal database table is provided by a database management system, and the at least one processor-readable program storage device is further configured with instructions for execution by the one or more processors for performing the operations comprising:

in response to selection of a first one of the plurality of methods,

for each object identifier in the file,

generating a character string array having in character string format, data from the vertical database table including, an object identifier and attribute values for attributes associated with the object identifier;

generating an SQL insert command from the contents of the character string array;

issuing the SQL insert command to the database management system; and

issuing an SQL commit command to the database management system.

23. The program storage medium of claim 22, wherein the instructions for generating a character string array include instructions for determining a position for storage of an attribute value in the character string array using a hash of the attribute name associated with the attribute value.

24. The program storage medium of claim 22, wherein the first one of the plurality of methods is selected in response to the number of rows in the vertical table being less than a first threshold and the number of columns in the horizontal table being less than a second threshold.

25. The program storage medium of claim 18, wherein access to the vertical database table and access to the horizontal database table are provided by a database management system, and the at least one processor-readable program storage device is further configured with instructions for execution by the one or more processors for performing the operations comprising:

in response to selection of a first one of the plurality of methods,

generating a single SQL command that selects data from the vertical table and inserts the data in the horizontal table for each object identifier in the vertical table and each column in the horizontal table; and

issuing the SQL command to the database management system.

26. The program storage medium of claim 25, wherein the at least one processor-readable program storage device is further configured with instructions for execution by the one or more processors for performing the operations comprising:

determining an approximate percentage of a total number of attribute values that will have an undefined value in the horizontal table;

wherein the first one of the plurality of methods is selected in response to the approximate percentage being less than a first threshold percentage, the number of columns in the horizontal table being less than a second threshold, and the number of rows in the vertical table being less than a third threshold.

27. The program storage medium of claim 26, wherein the at least one processor-readable program storage device is further configured with instructions for execution by the one or more processors for performing the operations comprising:

in response to selection of a second one of the plurality of methods,

for each object identifier in the file,

generating a character string array having in character string format, data from the vertical database table including, an object identifier and attribute values for attributes associated with the object identifier; and

writing data from the character string array to a flat text file; and

loading data from the flat text file into the horizontal database using a utility function of the database management system, wherein the utility function is adapted to read data from a flat text file; and

wherein the second one of the plurality of methods is selected in response to the number of rows in the vertical table being greater than a fourth threshold and the number of columns in the horizontal table being less than the second threshold; and

wherein the second one of the plurality of methods is selected in response to the number of columns in the horizontal table being less than the second threshold, the number of rows in the vertical table being in the range of a fifth threshold to the third threshold, and the approximate percentage being greater than the first threshold percentage.

28. The program storage medium of claim 18, wherein access to the vertical database table and access to the horizontal database table are provided by a database management system, and the at least one processor-readable program storage device is further configured with instructions for execution by the one or more processors for performing the operations comprising:

in response to selection of a first one of the plurality of methods,

for each object identifier in the file,

generating a character string array having in character string format, data from the vertical database table including, an object identifier and attribute values for attributes associated with the object identifier; and

writing data from the character string array to a flat text file; and

loading data form the flat text file into the horizontal database using a utility function of the database management system, wherein the utility function is adapted to read data from a flat text file;

in response to selection of a second one of the plurality of methods,

for each object identifier in the file,

generating a character string array having in character string format, data from the vertical database table including, an object identifier and attribute values for attributes associated with the object identifier;

generating an SQL insert command from the contents of the character string array;

issuing the SQL insert command to the database management system; and

issuing an SQL commit command to the database management system;

in response to selection of a third first one of the plurality of methods,

generating a single SQL command that selects data from the vertical table and inserts the data in the horizontal table for each object identifier in the vertical table and each column in the horizontal table; and

issuing the SQL command to the database management system.

29. The program storage medium of claim 28, wherein the at least one processor-readable program storage device is further configured with instructions for execution by the one or more processors for performing the operations comprising:

determining storage space requirements of the flat text file;

determining availability of sufficient retentive storage space based on the storage space requirements of for the flat text file;

selecting the first method in response to the number of columns being less than a first threshold, the number of rows being greater than a second threshold, and there being available sufficient retentive storage space for the flat text file.

30. The program storage medium of claim 29, wherein the at least one processor-readable program storage device is further configured with instructions for execution by the one or more processors for performing the operations comprising:

determining an approximate percentage of a total number of attribute values that will have an undefined value in the horizontal table;

in response insufficient retentive storage space being available for the flat text file,

selecting the second method in response to the number of rows being less than a third threshold or the approximate percentage being greater than a fourth threshold percentage;

selecting the third method in response to the number of rows being greater than or equal to the third threshold and the approximate percentage being less than or equal to the fourth threshold percentage.

APPENDIX OF EVIDENCE FOR APPLICATION NO. 10/814,715

Appellant is unaware of any evidence submitted in this application pursuant to 37 C.F.R. §§ 1.130, 1.131, and 1.132.

APPENDIX OF RELATED PROCEEDINGS FOR APPLICATION NO. 10/814,715

Appellant is unaware of any related appeals, interferences or judicial proceedings.